Listing of Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of determining whether a requested permission, wherein the permission is at least one of a set of permissions, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the method comprising:

associating a first permission grant object with a first code assembly in the runtime call stack;

dynamically overriding the set of permissions that is assigned to a second permission grant object associated with ~~at least one other~~ a second code assembly preceding the first code assembly;

creating a permission request object within the called code frame to demand the requested permission;

demanding via the permission request object the requested permission from the first permission grant object to allow the called code frame to perform the protected operation;

determining whether the requested permission is provided in association with the first code assembly by the first permission grant object, responsive to the demanding operation; and

permitting the execution of the called code frame to perform the protected operation, if the requested permission is provided in association with the first code assembly, whereby a full walk of the runtime call stack may be avoided.

2. (Original) The method of claim 1 wherein the called code frame is included within the first code assembly.

3. (Original) The method of claim 1 wherein the called code frame is included within a lower level code assembly following the first code assembly in the runtime call stack.

4. (Original) The method of claim 1 further comprising:

associating a second permission grant object with a second code assembly loaded in the runtime call stack, the second code assembly preceding the first code assembly in the runtime call stack.

5. (Previously Presented) The method of claim 4 further comprising:

determining whether the requested permission is provided in associating with the second code assembly by the second permission grant object.

6. (Previously Presented) The method of claim 1 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that a permission grant object associated with the at least one other code assembly preceding the first code assembly need not be evaluated to determine whether a specified permission is satisfied in association with the other code assembly in the runtime call stack, regardless of whether the specified permission is provided by the permission grant object associated with the other code assembly; and

permitting execution of the called code frame to perform the protected operation, if the requested permission is a subset of the specified permission.

7. (Previously Presented) The method of claim 1 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that a permission grant object associated with the at least one other code assembly preceding the first code assembly does not satisfy a specified permission within the runtime call stack, regardless of whether the specified permission is provided by the permission grant object associated with the other code assembly; and

preventing execution of the called code frame to perform the protected operation, if the requested permission is a subset of the specified permission.

8. (Previously Presented) The method of claim 1 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that a permission grant object associated with the at least one other code assembly preceding the first coded assembly satisfied only one or more specified permissions, regardless of whether one or more other permissions are also provided by the permission grant object associated with the other code assembly; and

permitting execution of the called code frame to perform the protected operation, only if the requested permission is a subset of the specified permissions.

9. (Original) The method of claim 1 wherein the permission object encoded in the code assembly and the corresponding permission object encoded in the permission grant object satisfy a common permission interface.

10. (Original) The method of claim 1 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual method of the first code assembly.

11. (Original) The method of claim 1 wherein the operation of associating a first permission grant object with a first code assembly comprises;

associating the first permission grant object with an individual class of the first code assembly.

12. (Original) The method of claim 1 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual module of the first code assembly.

13. (Previously Presented) A method determining whether a requested permission, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the method comprising:

associating a first permission grant object with a first code assembly in the runtime call stack;

associating a second permission grant object with a second code assembly in the runtime call stack;

computing a first intersection of permissions provided by the first permission grant object and the second permission grant object;

recording the first intersection of permissions to provide a cached permission intersection;

demanding the requested permission; and

permitting execution of the called code frame if the requested permission is a subset of the cached permission intersection, whereby a full walk of the runtime call stack may be avoided.

14. (Original) The method of claim 13 further comprising:

associating a third permission grant object with a third code assembly in the runtime call stack;

computing a second intersection of permissions provided by the first permission grant object, the second permission grant object, and third permission grant object; and

recording the second intersection of permissions to provide the cached permission intersection.

15. (Original) The method of claim 13 wherein the called code frame is included within the first code assembly.

16. (Original) The method of claim 13 wherein the called code frame is included within a lower level code assembly following the first code assembly in the runtime call stack.

17. (Original) The method of claim 13 wherein the operation of associating a first permission grant object with a first code assembly comprises;

associating the first permission grant object with an individual method of the first code assembly.

18. (Original) The method of claim 13 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual class of the first code assembly.

19. (Original) The method of claim 13 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual module of the first code assembly.

20. (Previously Presented) A runtime system for determining whether a requested permission of a set of permissions, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the runtime system comprising:

a first code assembly loaded into the runtime call stack;

a first permission grant object associated with the first code assembly comprising one or more permissions available to the first code assembly; and

a first permission request object created by the called code frame requesting the requested permission from the first permission grant object, wherein the called code frame is permitted to execute the protected operation if the first permission request object determines from the permission grant object that the requested permission is satisfied by the first code assembly and wherein the permission set is dynamically overridden to modify one permission within the set of permissions and whereby a full walk of the runtime call stack may be avoided.

21. (Original) The runtime system of claim 20 wherein the called code frame is included within the first code assembly.

22. (Original) The runtime system of claim 20 wherein the called code frame is included within a lower level code assembly following the first code assembly in the runtime call stack.

23. (Original) The runtime system of claim 20 wherein the first permission grant object is associated with an individual method of the first code assembly.

24. (Original) The runtime system of claim 20 wherein the first permission grant object is associated with an individual class of the first code assembly.

25. (Original) The runtime system of claim 20 wherein the first permission grant object is associated with an individual module of the first code assembly.

26. (Previously Presented) A runtime system for determining whether a requested permission, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the runtime system comprising:

a first permission grant object associated with a first code assembly in the runtime call stack;

a second permission grant object associated with a second code assembly in the runtime call stack; and

a cache storing an intersection of permissions provided by the first permission grant object and the second permission grant object, wherein execution of the called code frame is permitted and a full walk of the runtime call stack may be avoided if the requested permission is a subset of the cached permission intersection.

27. (Original) The runtime system of claim 26 wherein the intersection is computed and stored in the cache before the requested permission is requested.

28. (Previously Presented) A computer program product encoding a computer program for determining whether a requested permission, wherein the permission is at least one of a set of permissions, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the computer process comprising:

associating a permission grant object with a first code assembly in the runtime call stack;

dynamically overriding the set of permissions that is assigned to the assembly associated with at least one other code assembly preceding the first code assembly;

creating a permission request object within the called code frame to demand the requested permission;

demanding via the permission request object the requested permission from the permission grant object to allow the called code frame to perform the protected operation;

determining whether the requested permission is provided in association with the first code assembly by the permission grant object, responsive to the demanding operation; and

permitting execution of the called code frame to perform the protected operation whereby a full walk of the runtime call stack may be avoided, if the requested permission is provided in association with the first code assembly.

29. (Original) The computer program product of claim 28 wherein the called code frame is included within the first code assembly.

30. (Original) The computer program product of claim 28 wherein the called code frame is included within a lower level code assembly following the first code assembly in the runtime call stack.

31. (Original) The computer program produce of claim 28 wherein the computer process further comprises:

associating a second permission grant object with a second code assembly loaded in the runtime call stack, the second code assembly preceding the first code assembly in the runtime call stack.

32. (Original) The computer program product of claim 31 wherein the computer process further comprises;

determining whether the requested permission is provided in association with the second code assembly by the second permission grant object.

33. (Previously Presented) The computer program produce of claim 28 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that a permission grant object associated with at least one other code assembly preceding the first code assembly need not be evaluated to determine whether a specified permission is satisfied in association with the other code assembly in the runtime call stack, regardless of whether the specified permission is provided by the permission grant object associated with the other code assembly; and

permitting execution of the called code frame to perform the protected operation, if the requested permission is a subset of the specified permission.

34. (Previously Presented) The computer program produce of claim 28 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that a permission grant object associated with the at least one other code assembly preceding the first code assembly does not satisfy a specified permission within the runtime call stack, regardless of whether the specified permission is provided by the permission grant object associated with the other code assembly; and

preventing execution of the called code frame to perform the protected operation, if the requested permission is a subset of the specified permission.,

35. (Previously Presented) The computer program product of claim 28 wherein the operation of dynamically overriding comprises:

asserting within the first code assembly that permission grant object associated with the at least one other code assembly preceding the first code assembly satisfies only one or more specified permissions, regardless of whether one or more other permissions are also provided by the permission grant object associated with the other code assembly; and

permitting execution of the called code frame to perform the protected operation, only if the requested permission is a subset of the specified permissions.

36. (Original) The computer process of claim 28 wherein the permission object encoded in the code assembly and the corresponding permission object encoded in the permission grant object satisfy a common permission interface.

37. (Original) The computer program produce of claim 28 wherein the operation of associating a permission grant object with a first code assembly comprises;

associating the first permission grant object with an individual method of the first code assembly.

38. (Original) The computer program product of claim 28 wherein the operation of associating a permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual class of the first code assembly.

39. (Original) The computer program product of claim 28 wherein the operation of associating a permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual module of the first code assembly.

40. (Previously Presented) A computer program product encoding a computer program for determining whether a requested permission, requested by a called code frame, is satisfied within a runtime call stack so as to allow the called code frame to perform a protected operation, the computer process comprising:

associating a first permission grant object with a first code assembly in the runtime call stack;

associating a second permission grant object with a second code assembly in the runtime call stack;

computing a first intersection of permissions provided by the first permission grant object and the second permission grant object;

recording the first intersection of permissions to provide a cached permission intersection; and

permitting execution of the called code frame if the requested permission is a subset of the cached permission intersection, whereby a full walk of the runtime call stack may be avoided.

41. (Original) The computer program product of claim 40 wherein the computer process further comprises:

associating a third permission grant object with a third code assembly in the runtime call stack;

computing a second intersection of permissions provided by the first permission grant object, the second permission grant object, and the third permission grant object; and

recording the second intersection of permissions to provide the cached permission intersection.

42. (Original) The computer program product of claim 40 wherein the called code frame is included within the first code assembly.

43. (Original) The computer program product of claim 40 wherein the called code frame is included within a lower level code assembly following the first code assembly in the runtime call stack.

44. (Original) The computer program product of claim 40 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual method of the first code assembly.

45. (Original) The computer program produce of claim 40 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual class of the first code assembly.

46. (Original) The computer program product of claim 40 wherein the operation of associating a first permission grant object with a first code assembly comprises:

associating the first permission grant object with an individual module of the first code assembly.